Aprendizaje automático relacional (Predicción de una banda de Londres)

José Manuel Tabares Rodríguez
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
jostabrod@alum.us.es / jmtr2000@gmail.com

Víctor Monteseirín Puig
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
vicmonpui@alum.us.es / victormp00@gmail.com

***Resumen*—Para este trabajo, nuestro principal objetivo ha sido abordar un problema de clasificación relacional, aprendiendo a elaborar una función que dados los atributos de un nodo de una red, este sea capaz de clasificar correctamente dicho nodo entre una serie de clases. Para ello, se deberá usar una función que divida el conjunto de datos en un conjunto de entrenamiento, y que se pruebe con algunos nodos de prueba. Todo mediante los métodos de la librería sklearn, que dispone de una gran ayuda para implementar estas funciones.**

Para ello, se usarán métodos propios de grafos escogidos a partir de la librería de networkx[1], también se deberá aprender a usar algunos atributos relacionales de los grafos como la centralidad, el coeficiente de clustering o el grado de entrada y salida de cada nodo.

El resultado final de este trabajo ha concluido en el estudio de la clasificación de diferentes modelos como naive bayes o k-vecinos más cercanos sobre un grafo de los participantes de un secuestro y sus respectivos atributos, como conclusión se puede decir que la información resultante es útil para conocer en un caso hipotético, si nuevos miembros de la banda iría o no a prisión.

I. Introducción

Las bandas y pandillas criminales son un rasgo muy común en la sociedad y aunque lo ideal seria evitar la creación de estas mismas, tenemos claro que el mundo no es un lugar de ensueño. Por esta misma razón la inteligencia artificial pretende buscar patrones asociados a los miembros de estas organizaciones para predecir sus posibles pasos o acciones a partir de los datos de los que se dispone de los mismos, por ello, se ha decidido estudiar un conjunto de datos de una pandilla de Londres[2] que se explicará más adelante, y con esto poder predecir sus acciones, así como su desarrollo.

En este trabajo en concreto vamos a realizar una investigación acerca de una importante pandilla de Londres en el periodo entre 2005 y 2009, mediante datos sacados de informes de arrestos y condenas policiales de "todos los miembros confirmados" de la banda. Los datos de los que disponemos son edad, lugar de nacimiento, residencia, arrestos, condenas, si fueron a prisión, si escuchaban música, además de las relaciones entre ellos que son desde pasar el rato, cometer delitos o ser parientes.

Se va a estudiar quiénes de dichas personas involucradas en la banda acabará estando en prisión. Teniendo como objetivo saber que características necesitas tener y que tan relacionado con los otros involucrados necesita estar para acabar en prisión e intentar conseguir un método eficaz para predecir si un nuevo miembro de la banda que cumpla ciertas relaciones con ellos sería propensa a acabar en prisión.

Se dispone para este trabajo de dos archivos csv, uno dispone de una matriz de adyacencia de 54x54 que representa las relaciones entre los nodos(personas pertenecientes a la banda de Londres), y también de un archivo csv con los atributos de cada nodo.

Este trabajo está organizado siguiendo la siguiente estructura: Inicialmente en la sección "Preliminares" se mostrarán los métodos empleados para resolver los problemas enfrentados en el trabajo. En la sección "Metodología" se describirá el método implementado en el trabajo, además esta dispondrá de imágenes y pseudocódigo, posteriormente en "Resultados" se explicarán los experimentos realizados, los resultados obtenidos y se analizarán dichos resultados para que en "Conclusiones", se proporcione un resumen de los datos aprendidos y una reflexión acerca de los mismos. Por último, mencionar que se añadirán todas las referencias a las fuentes utilizadas para encontrar información en la sección de "Referencias" al final del documento.

II. Preliminares

Aquí haremos un breve resumen a modo de introducción de las técnicas empleadas para la resolución del problema propuesto.

A. Modelos y técnicas empleadas

Se pueden usar listas por puntos como sigue:

- Validación Cruzada: es una técnica usada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. En este trabajo se usa esta técnica para calcular diferentes medidas de evaluación sobre diferentes particiones para así obtener un conjunto de datos y modelo adecuado.
- Naive Bayes: se trata de un simple, pero efectivo clasificador probabilístico fundamentado en el teorema de Bayes, este supone independencia entre las variables predictoras, por esto mismo recibe el nombre de naive(ingenuo), para el caso de este trabajo en concreto

se usará naive bayes para estudiar si los nodos del grafo creado(miembros de la banda) fueron a prisión.

- KNN: El método de KNN o k-vecinos más próximos nos servirá para construir un modelo probabilístico a partir de los ejemplos que dispone el modelo de datos. Clasificaremos los nuevos ejemplos en función de las categorías de los ejemplos más cercanos, por tanto hay que manejar distancia entre los mismos, para esta labor hemos utilizado la distancia de Hamming(basada en el número de componentes en los que se difiere) y la euclídea(usada mediante la fórmula de la distancia euclídea), como objetivo final compararemos cuál es la que mejor se adecua.
- Árbol de decisión: Se trata de uno de los algoritmos de clasificación más útiles y visuales donde el objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas de decisión simples extraídas de los atributos de los datos. Además gracias a la librería sklearn[3] se puede exportar el árbol de decisión resultante, lo que proporciona un grado en el que se puede distinguir de manera visible ciertos atributos relacionados con los datos con los que se trabaja. El árbol está formado por nodos interiores que en este caso son los atributos, arcos, que son los posibles valores del nodo del que se origina y hojas, que tienen asignadas los valores resultantes. Es destacable que cuanto más profundo es el árbol, más complejas son las reglas de decisión usadas y más ajustado será el modelo, aunque hay que trabajar con cautela debido a que la creación de árboles demasiado complejos puede provocar una no correcta generalización de los datos y la posibilidad de causar sobreajuste[8].

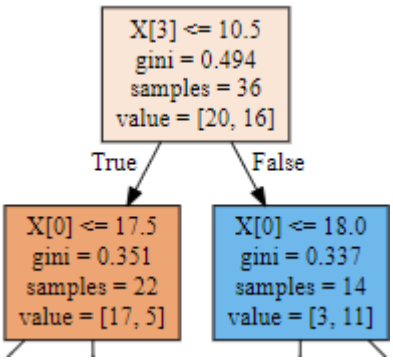
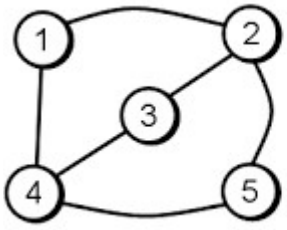


Fig. 1. Árbol de Decisiones recortado

B. Datos usados

- Matriz de adyacencia: Se trata de una matriz cuadrada de n filas x n columnas siendo n el número de nodos del grafo, cada celda de la matriz puede tener valores del 0 al 4, que representan la relación entre los diferentes miembros de la banda, ”0” representa no relación entre los miembros, ”1” representa que estos compartían su tiempo,”2” implica que los delincuentes cometían fechorías menores,”3” tiene que ver con la ejecución de delitos más graves y ”4” significa que cometen delitos graves o que los miembros sean parientes. Estos últimos datos serían los pesos del grafo, aunque en este trabajo no se trabaja con ellos.



M	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	1	0	1	0	1
5	0	1	0	1	0

Fig. 2. Ejemplo de matriz de adyacencia

- Edad: Es un atributo que marca los años que tienen los diferentes miembros de la banda.
- Lugar de nacimiento: Es un atributo que muestra los posibles lugares codificados con números del 1 al 4 en los que ha nacido cada miembro de la banda, estos lugares son África del Oeste codificado como 1, El Caribe como 2, Gran Bretaña como 3 y África del Este como 4.
- Residencia: Este atributo da a entender si el miembro en cuestión dispone o no de una residencia particular.
- Arrestos: Este atributo muestra si el miembro de la banda ha sido arrestado y en caso de que haya sido arrestado el número de veces que el miembro en cuestión ha sido arrestado.
- Condenas: Este atributo indica si el miembro de la banda ha sido condenado y en caso de que haya sido condenado el número de condenas a las que ha sido sometido el miembro en cuestión.
- Prisión: Este atributo indica si el miembro de la banda en cuestión ha estado o no en prisión.
- Música: Se trata de otro atributo que presentan los miembros de la banda que representa si estos tienen relación o no con la música. Se cree que escuchar música triste o feliz puede poner a las personas en un estado de ánimo diferente, pero también cambiar lo que las personas notan, en cualquier caso puede ser conveniente estudiar si la relación entre música y ”gánsteres” puede tener repercusión en otros atributos de su vida.

Age	Birthplace	Residence	Arrests	Convictions	Prison	Music	Ranking
20	1	0	16	4	1	1	1
20	2	0	16	7	1	0	2
19	2	0	12	4	1	0	2
21	2	0	8	1	0	0	2
24	2	0	11	3	0	0	2

Fig. 3. Tabla de atributos con las 5 primeras filas

- Ranking: Por último el atributo de ”ranking” nos indicará la posición del individuo dentro de la banda, siendo los posibles valores desde el ”1” hasta el ”5”, aquellos individuos clasificados como ”5” serían aquellos con

mejor posición dentro de la banda y aquellos con ”1” los peores.

C. Métricas relacionales utilizadas

- Grado: También llamado valencia de un nodo, es el número de aristas (arcos) que inciden sobre el nodo. En el caso particular de este trabajo el grado de un miembro indica el número de miembros con los que este se relaciona.
- Centralidad: Se trata de un atributo relacional que realiza una medida estimada de un nodo dentro de un grafo y gracias a esta medida se puede conocer la importancia o relevancia del nodo en ese grafo. Conocer la centralidad de un nodo puede ayudar a determinar el impacto que este causa dentro del conjunto del que forma parte.
- Clustering: El grado de clustering de un nodo en un grafo cuantifica que tanto está interconectado con sus vecinos, siendo estos los vértices conectados al primero. Se puede observar un ejemplo en la Fig. 4

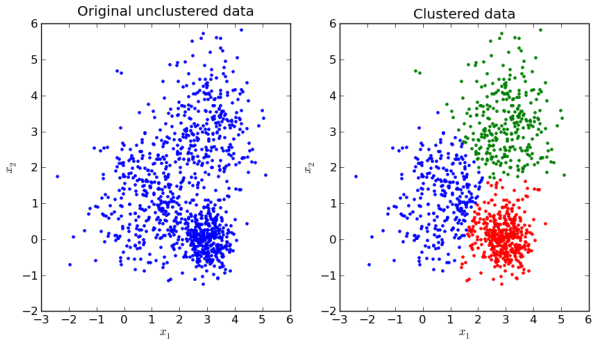


Fig. 4. Ejemplo de Clustering

- Coloreado: El coloreado de grafos es una técnica de etiquetado de grafos consistente en asignar unas etiquetas llamadas colores a los diferentes elementos de los grafos (en este caso se le asignan a los nodos). Se debe hacer de tal forma que no puedan existir vértices adyacentes que dispongan del mismo color. Tal y como es mostrado en la Fig. 6 que se puede observar un grafo coloreado.

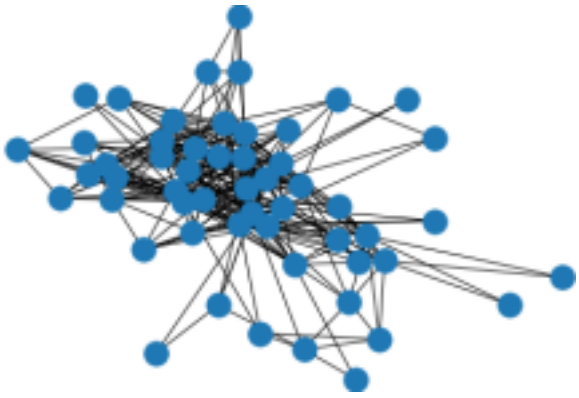


Fig. 5. Grafo de la banda de Londres

III. Metodología

Esta sección va a ser dedicada a una descripción completa del método y demás funciones implementados durante el transcurso del trabajo, para ello se proporcionarán descripciones amplias, capturas de resultados, pseudocódigos, esquemas, tablas, etc.

1) *Creación del grafo:* Utilizando la librería pandas, que es una librería especializada en el manejo y análisis de estructuras de datos[4], se leen los datos de la matriz de adyacencia encontrada en el archivo .csv llamado ”LONDON GANG”, a continuación se borrará la primera fila de esta para que la función de lectura no de error, este ocurre dado que la primera fila se trata de una cabecera que muestra números representando los nodos del grafo, así como el propio grafo dibujado el cual se dibuja utilizando la librería networkx[1]. El grafo se puede observar en la Fig. 5
2) *Obtención de los datos:* A continuación se usará el archivo ”LONDON GANGS ATTR ” que contiene los atributos de cada nodo del grafo, guardaremos sus datos utilizando nuevamente la librería pandas[4].
3) *Posteriormente se procederá a obtener los atributos relacionales:*
   a) *Coloreado:* Primeramente se obtendrá el coloreado del grafo llamando a la librería networkx[1], que devolverá un diccionario (Tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor), cuyas claves serán los nodos y los valores del color asociado a cada nodo (indicado con un número, cada número representa un color distinto), esto será transformado en una lista de tuplas usando la función item() de los diccionarios(), posteriormente dicha lista se ordenará para que finalmente, se introduzcan los colores ya ordenados en una lista. Con esta lista ya podremos agregar el apartado de coloreado en la tabla que corresponderá a este atributo relacional.

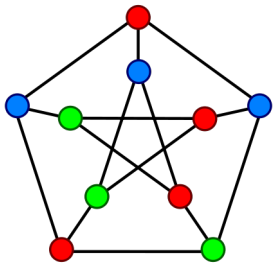


Fig. 6. Ejemplo de un grafo al que se le aplica coloreado

   b) *Grado:* Para obtener el grado de cada nodo del grafo, se ha llamado a la función degree que devuelve una lista de tuplas ordenadas por el primer elemento de las mismas. Cada tupla dispone de dos elementos, el primero representa al nodo y el segundo representa la valencia asociada a dicho

nodo. Al venir ordenada se realiza un bucle para recoger únicamente el segundo elemento de las tuplas y recogerlos en una lista, esta lista será introducida en la tabla de atributos del grafo como otro nuevo atributo relacional.

c) *Centralidad:* Nuevamente se llama a la librería networkx[1] con la que se procede a calcular la centralidad del grafo llamando a la función ”degree-centrality”, esto devuelve un diccionario donde las claves son los distintos nodos del grafo y los valores serán la centralidad de los mismos. A continuación se usa un bucle para convertir los valores en una lista que será añadida a la tabla como atributo relacional ”Centralidad”.

d) *Clustering:* Obtendremos el clustering del grafo llamando a la librería networkx[1],El resultado de esto es un diccionario que indica el miembro de la banda y su grado de clustering. Luego introduciremos el clustering de cada miembro en una lista. Esta lista será usada para crear el atributo relacional Clustering que será posteriormente añadido a la tabla con todos los atributos.

e) *Coloreado:* Obtendremos el coloreado del grafo llamando a la librería networkx[1] y utilizando la función ”greedy-color”, El resultado de esto es un diccionario que indica el color de cada miembro de la banda. Esto tendremos que ordenarlo por valor para poder conseguir mediante un bucle, una lista ordenada por nodos de todos los colores del grafo.

En la Fig. 7 se pueden observar los atributos relacionales en formato de tabla.

grado	Centrality	Clustering	Coloreado
25	0.471698	0.453333	0
22	0.415094	0.545455	6

Fig. 7. Sección de la tabla de las métricas relacionales

4) *Modelos de entrenamiento:* Para la elaboración de todos los modelos que se explicarán posteriormente, es necesario explicar el cómo se obtienen los atributos y objetivos de entrenamiento y prueba. Estos atributos, como su propio nombre indican, son aquellos que se encargan de entrenar y probar los diferentes modelos con el objetivo de estimar su eficacia real. Para obtener los datos mencionados se ha utilizado el paquete ”model-selection”[5] de la librería sklearn, el cual dispone de la función ”train-test-split” que devuelve los valores citados a partir de los atributos y objetivo que se le pasa por parámetros(en este caso los atributos propios del grafo y los relacionales explicados anteriormente), alternando estos atributos se pueden conseguir resultados distintos que permitirán un mejor estudio de los modelos en general.

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

- P(A): Probabilidad de A
- P(R|A): Probabilidad de que se de R dado A
- P(R): Probabilidad de R
- P(A|R): Probabilidad posterior de que se de A dado R

Fig. 8. Fórmula de calcular la probabilidad ‘posterior’ de Naive bayes

A continuación se explica cuál ha sido la metodología utilizada para desarrollar los diferentes modelos durante el trabajo:

a) *Naive Bayes:* La metodología seguida para este modelo es algo simple, no se le ha dedicado demasiado énfasis debido a que este está pensado para atributos categóricos, de forma que enfocarlo con atributos relacionales no es tan interesante. Por tanto, lo único que presenta esta parte del trabajo es el método entrenado con los atributos originales y sin los relacionales, y la parte correspondiente al cálculo de la puntuación del mismo, esta última se obtiene mediante la función ”score” importada del mismo lugar del método de ”naive bayes” al que además es importante comentar que se la ha aplicado una k(parámetro de suavizado), correspondiente a 1.

b) *K-Vecinos:* La estrategia seguida para desarrollar este modelo ha sido utilizar el paquete neighbors[6] de la librería sklearn, al cual se le pasa por parámetros el número de vecinos y el tipo de distancia que se usa para deducir como está más cerca(en este caso hamming y euclídea), a esto se le aplicará validación cruzada(técnica importada de la función ”cross-val-score”[7] de la librería sklearn) para determinar diversas puntuaciones del modelo intercalando atributos y así comparar entre los posibles resultados y escoger uno para el resultado final. Además mediante un bucle podemos comparar la puntuación del método con diferentes vecinos, en este caso se compara desde 0 hasta 10 vecinos.

Fig. 9. Metodo K vecinos más próximos

c) *Árboles de Decisión:* La estrategia utilizada para este método ha sido utilizar la función tree.DecisionTreeClassifier[8] importada de la librería sklearn para aplicar el algoritmo de Clasificación por árboles de decisión, este será entrenado por los diferentes atributos de entrenamiento y probado mediante validación cruzada, tras esto se puede comparar con qué tipos de atributos es más óptimo este modelo. Además este modelo permite dibujar un grafo del árbol generado lo que permite una comparación visual más óptima, para poder aprovechar esta característica mejor hemos exportado el grafo para que así sea más manejable.

Euclídea: $d_e(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Fig. 10. Formula de la distancia Euclídea

IV. Resultados

Resultados de los siguientes experimentos:

- Resultados sobre el experimento del modelo naive bayes: Siguiendo la metodología explicada en el apartado de naive bayes y mediante los atributos de entrenamiento y pruebas (sin la parte relacional) se ha conseguido obtener una puntuación correspondiente a 0,61112 para el modelo comentado.

```
Puntuación de naive bayes: 0.611
```

Fig. 11. Resultado de Naive Bayes

- Resultados sobre el experimento de KNN(K-vecinos más cercanos):A continuación se pueden observar los resultados de aplicar la metodología del modelo de KNN (vecinos más próximos) usando los 2 tipos de distancias mencionadas anteriormente para definir la cercanía, estos resultados nos proveen la media de valores de puntuación obtenidos mediante validación cruzada probados para los atributos originales, y otros atributos relacionales, además también se pueden observar estos resultados alternando entre distintos números de vecinos y distintas distancias. En las Fig. 12 ,Fig. 13 ,Fig. 14 y Fig. 15 se pueden observar los diferentes resultados de KNN.
- Resultados sobre el experimento de Árboles de decisión: aplicando el algoritmo de árboles de decisión y entrenando mediante diferentes conjuntos de entrenamiento, además de aplicarle validación cruzada, se han conseguido obtener los siguientes resultados indicando la media de puntuación del modelo entre los distintos resultados de la validación cruzada. En la Fig. 16 se pueden observar dichos resultados.

Ahora se exponen las comparaciones y conclusiones sobre los experimentos anteriores:

- La primera comparación a realizar será entre los resultados del modelo de KNN, compararemos sus distintas

```
Para los atributos originales la puntuación
con 1 vecino es 0.5333333333333333
Para los atributos originales la puntuación
con 2 vecinos es 0.5666666666666667
Para los atributos originales la puntuación
con 3 vecinos es 0.4666666666666666
Para los atributos originales la puntuación
con 4 vecinos es 0.4916666666666667
```

Fig. 12. Resultado de KNN usando distancia Hamming

```
Para los atributos originales la media
con 1 vecino es 0.7750000000000001
Para los atributos originales la media
con 2 vecinos es 0.7250000000000001
Para los atributos originales la media
con 3 vecinos es 0.6083333333333333
Para los atributos originales la media
con 4 vecinos es 0.7583333333333334
```

Fig. 13. Resultados de KNN usando distancia Euclídea

```
media de la puntuación con 1 vecino y distancia hamming:
Para todos los atributos relacionales: 0.4416666666666667
Para el atributo relacional grado:  0.475
Para el atributo relacional centralidad:  0.475
Para el atributo relacional clustering:  0.5333333333333333
Para el atributo relacional coloreado  0.5583333333333333
============================================================
media de la puntuación con 2 vecinos y distancia hamming:
Para todos los atributos relacionales: 0.6500000000000001
Para el atributo relacional grado:  0.5166666666666667
Para el atributo relacional centralidad:  0.5166666666666667
Para el atributo relacional clustering:  0.5666666666666667
Para el atributo relacional coloreado  0.5916666666666667
============================================================
media de la puntuación con 3 vecinos y distancia hamming:
Para todos los atributos relacionales: 0.5333333333333334
Para el atributo relacional grado:  0.4916666666666666
Para el atributo relacional centralidad:  0.4916666666666666
Para el atributo relacional clustering:  0.4666666666666666
Para el atributo relacional coloreado  0.5166666666666667
============================================================
media de la puntuación con 4 vecinos y distancia hamming:
Para todos los atributos relacionales: 0.5416666666666667
Para el atributo relacional grado:  0.4916666666666667
Para el atributo relacional centralidad:  0.4916666666666667
Para el atributo relacional clustering:  0.4916666666666667
Para el atributo relacional coloreado  0.4916666666666667
```

Fig. 14. Resultados de KNN usando datos relacionales y distancia Hamming

medias en función a los atributos, vecinos y distancias. Observando las figuras en las que se muestran los resultados, se puede afirmar que en general para KNN es más eficiente la distancia euclídea que la distancia de Hamming porque las medias del primero son en general mucho más elevadas que las del segundo. A continuación se compara el modelo para diferentes vecinos y atributos. Expresado esto se puede afirmar que para 1 vecino es conveniente usar la media de los atributos originales, esto es debido a que al usar atributos relacionales se observa que la media baja independientemente de usarlos juntos o separados, por tanto la media más alta para 1 vecino es

```
media de la puntuación con 1 vecino y distancia euclídea:
Para todos los atributos relacionales: 0.6916666666666667
Para el atributo relacional grado:  0.6666666666666667
Para el atributo relacional centralidad:  0.7083333333333333
Para el atributo relacional clustering:  0.6833333333333333
Para el atributo relacional coloreado  0.6583333333333333
============================================================
media de la puntuación con 2 vecinos y distancia euclídea:
Para todos los atributos relacionales: 0.7
Para el atributo relacional grado:  0.7083333333333334
Para el atributo relacional centralidad:  0.7250000000000001
Para el atributo relacional clustering:  0.7000000000000001
Para el atributo relacional coloreado  0.6083333333333333
============================================================
media de la puntuación con 3 vecinos y distancia euclídea:
Para todos los atributos relacionales: 0.6916666666666667
Para el atributo relacional grado:  0.6916666666666667
Para el atributo relacional centralidad:  0.6083333333333333
Para el atributo relacional clustering:  0.5833333333333333
Para el atributo relacional coloreado  0.6333333333333333
============================================================
media de la puntuación con 4 vecinos y distancia euclídea:
Para todos los atributos relacionales: 0.7333333333333334
Para el atributo relacional grado:  0.725
Para el atributo relacional centralidad:  0.7583333333333334
Para el atributo relacional clustering:  0.7583333333333334
Para el atributo relacional coloreado  0.6916666666666667
```

Fig. 15. Resultados de KNN con datos relaccionales y distancia Euclídea

```
la media de la puntuación es:
Para los atributos originales  0.5979166666666667
Para los atributos relacionales  0.6597916666666667
Para el atributo relacional grado 0.725
Para el atributo relacional centralidad  0.6665979166666667
Para el atributo relacional clustering  0.7583333333333334
Para el atributo relacional coloreado  0.5374993124999999
```

Fig. 16. Resultados para árboles de decisión usando datos relacionales

”0.775”. Para 2 vecinos, se puede observar que la mejor estadística de puntuación es de nuevo la de los atributos categóricos originales, aunque esta sea la misma que la que presenta el atributo ”centralidad”, se considera mejor porque requiere de un menor número de atributos, por tanto la media más alta para 2 vecinos es ”0.725”. Para 3 vecinos, se cumple que el mejor resultado entre las medias de puntuación es la que presenta el atributo grado, aunque el conjunto con todos los atributos relacionales también dispone de la misma media, se considera mejor dado que requiere de menos atributos para proporcionar un buen valor. Por tanto la media más alta para 3 vecinos es ”0.691”. Para 4 vecinos, nuevamente se afirma que la mejor de las estadísticas es la que contiene los atributos originales, porque de nuevo contiene la misma puntuación que los demás conjuntos y menos atributos, por tanto la media más alta para 4 vecinos es ”0.758”. La conclusión obtenida a raíz de estos resultados sobre el modelo KNN es que aumentar la cantidad de vecinos usados no necesariamente aumenta la calidad del modelo, esto puede deberse a causas de sobreajuste[8], sin embargo se puede afirmar que, generalmente es más efectivo usar en este método atributos no relacionales propios del grafo salvo en determinados casos, porque en determinadas circunstancias usar muchos atributos también puede causar sobreajuste[8]. Por último se declara que el mejor caso para usar KNN es usar solo los atributos originales, distancia euclídea y un único vecino, esto provee una estadística de ”0.775”.

- En este segundo experimento compararemos la eficacia del modelo de árbol de decisión. Esto ha sido medido comparando el resultado de la validación cruzada del algoritmo que ofrece sklearn con los distintos conjuntos de atributos creados anteriormente. Se puede comprobar que es mejor utilizar atributos relacionales a usar meramente los atributos originales, no obstante hay que destacar que no todos son igual de importantes, es más, se deduce que lo más efectivo es utilizar únicamente los originales más un último atributo que marque el grado de ”clusterización”, este en concreto sería el más efectivo del modelo con un valor de 0.7583.

Como conclusión final se van a comparar los resultados obtenidos de los 3 modelos, se puede explicar que el modelo de ”naive bayes” es el peor entre todos, el mejor caso de modelo de árboles de decisión es parecido al mejor caso de modelo de KNN, pero este último es mejor por décimas. Por último se afirma que el modelo de KNN sin los atributos relacionales, en este caso, sería el más eficiente para saber si nuevos miembros de la banda irán o no a prisión.

V. Conclusiones

En esta última sección se van a explicar las impresiones obtenidas durante el trabajo.

En primer lugar es importante decir que durante todo el transcurso del mismo, se ha proporcionado una libertad extraordinaria para la búsqueda de información que se ha necesitado, aunque también hay que decir que se han proporcionado unos primeros pasos para empezar lo cual es realmente digno de mención dado que estas nociones como el uso de kaggle o de librerías ayudan al alumnado a entender mejor el trabajo en general.

En segundo lugar, se puede destacar que se han conseguido nuevos conocimientos probando los métodos y funciones de las librerías usadas, por ejemplo networkx y scikit-learn.

Como ideas de mejora, los miembros de este grupo de trabajo en concreto Sugeriríamos un seguimiento obligatorio de los trabajos con objeto de saber si estos se están realizando correctamente, aunque el trabajo en general y la forma de realizarse nos parece adecuado.

Referencias

[1] Networkx documentation recuperado en https://networkx.org/

[2] The Colorado Index of Complex Networks (ICON) sacado de https://icon.colorado.edu, UcinetSoftware datasets london gangs https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/londongang

[3] Scikit-learn Machine Learning in Python recuperado en https://networkx.org/

[4] Aprende con Alf Recursos Educativos Libres manual de python La librería Pandas extraído de: https://aprendeconalf.es/docencia/python/manual/pandas/
[5] Scikit-learn Machine Learning in Python API recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
[6] Scikit-learn Machine Learning in Python Nearest Neighbors extraído de https://scikit-learn.org/stable/modules/neighbors.html#:~:text=NearestNeighbors%20implements%20unsupervised%20nearest%20neighbors,pairwise%20
[7] Scikit-learn Machine Learning in Python fuente en https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html
[8] Tema 2-Aprendizaje automático, CS.us.es, Universidad de Sevilla